AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for transparently optimizing data access, comprising:

gathering information related to data usage when a system is processing using a client runtime; and

determining a usage pattern of the system using gathered information,

wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the system.

2. (Currently Amended) The method of claim 1, further comprising:

pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in ~~determined by~~ the usage pattern of the system;

caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects locally in a cache associated with the system;

accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects from the cache; and

synchronizing the cache[[d]] ~~data~~ with a persistent data store, wherein the persistent data store stores the plurality of persistent objects.

3. (Original) The method of claim 1, further comprising:

generating a description of a business process model by retaining the usage pattern over a period of at least one execution of the system.

4. (Original) The method of claim 1, further comprising:

generating tests for the system by retaining the usage pattern over a period of at least one execution of the system.

5. (Original) The method of claim 1, wherein the usage pattern comprises pieces of information used together based on a relationship.

6. (Original) The method of claim 5, wherein the relationship is temporal.

7. (Original) The method of claim 5, wherein the relationship is causal.

8. (Original) The method of claim 1, further comprising:
deriving an initial usage pattern from application code analysis.

9. (Original) The method of claim 1, further comprising:
deriving an initial usage pattern from an empty set.

10. (Previously Presented) The method of claim 1, further comprising:
deriving an initial usage pattern from a specification of the system.

11. (Currently Amended) The method of claim 1, further comprising:
displaying the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in the usage pattern [[to]] on a display device.

12. (Original) The method of claim 1, further comprising:
generating documentation from the usage pattern.

13. (Currently Amended) A method for transparently optimizing data access, comprising:
gathering information related to data usage when a system is processing using a client runtime;
determining a usage pattern of the system using gathered information, wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the system;
pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in ~~determined by~~ the usage pattern of the system;

caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects locally in a cache associated with the system;

accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects from the cache; and

synchronizing the cache[[d]] ~~data~~ with a persistent data store, wherein the persistent data store stores the plurality of persistent objects.

14. (Currently Amended) A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:

gathering information related to data usage on the client-side when the distributed application is processing using a client runtime; and

determining a usage pattern using gathered information,

wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the client-side.

15. (Currently Amended) The method of claim 14, further comprising:

pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects ~~determined by~~ from the server-side using the usage pattern and a server runtime;

caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects on the client-side in a cache associated with the client-side;

accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects on the client-side using the cache; and

synchronizing the cache[[d]] ~~data~~ on the client-side with a persistent data store on the server-side, wherein the persistent data store stores the plurality of persistent objects.

16. (Original) The method of claim 14, wherein the usage pattern comprises pieces of information used together based on a relationship.

17. (Original) The method of claim 16, wherein the relationship is temporal.

18. (Original) The method of claim 16, wherein the relationship is causal.

19. (Original) The method of claim 14, wherein the data represents objects.

20. (Original) The method of claim 14, further comprising:
deriving an initial usage pattern from application code analysis.

21. (Original) The method of claim 14, further comprising:
deriving an initial usage pattern from an empty set.

22. (Currently Amended) The method of claim 14, further comprising:
deriving an initial usage pattern from a specification of the system.

23. (Currently Amended) The method of claim 14, further comprising:
displaying the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in the usage pattern ~~to~~ on a display device.

24. (Original) The method of claim 14, further comprising:
generating documentation from the usage pattern.

25. (Currently Amended) A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:

gathering information related to data usage on the client-side when the distributed application is processing using a client runtime;

determining a usage pattern using gathered information, wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the client-side;

pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects from the server-side using the usage pattern and a server runtime;

caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects on the client-side in a cache associated with the client-side;

accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects on the client-side using the cache; and

synchronizing the cache[[d]] ~~data~~ on the client-side with the persistent data store on the server-side, wherein the persistent data store stores the plurality of persistent objects.

26. (Currently Amended) A computer-readable medium having recorded thereon instructions executable by processing, the instructions for:

gathering information related to data usage when a system is processing using a client runtime;

determining a usage pattern of the system using gathered information, wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the system;

pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in ~~determined by~~ the usage pattern of the system;

caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects locally in a cache associated with the system;

accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects from the cache; and

synchronizing the cache[[d]] ~~data~~ with a persistent data store, wherein the persistent data store stores the plurality of persistent objects, wherein the persistent data store stores the plurality of persistent objects.

27. (Currently Amended) An apparatus for transparently optimizing data access, comprising:

means for gathering information related to data usage when a system is processing using a client runtime;

means for determining a usage pattern of the system using gathered information, wherein the usage pattern specifies at least one of a plurality of persistent objects and at least one of a plurality of attributes within the at least one of the plurality of persistent objects used by the system;

means for pre-fetching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects specified in ~~determined by~~ the usage pattern of the system;

means for caching ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects locally in a cache associated with the system;

means for accessing ~~data~~ the at least one of the plurality of persistent objects and at least one of the plurality of attributes within the at least one of the plurality of persistent objects from the cache; and

means for synchronizing the cache[[d]] ~~data~~ with a persistent data store, wherein the persistent data store stores the plurality of persistent objects.